

# XSLT *Jumpstarter*

Extracted from:

Level the Learning Curve and Put Your XML to Work

Copyright © 2015 Peloria Press.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.



Raleigh, North Carolina



# XSLT *Jumpstarter*

Level the Learning Curve  
and Put Your XML to Work

David James Kelly

Foreword by Dave Thomas

*The Martian language, as I have said, is extremely simple, and in a week I could make all my wants known and understand nearly everything that was said to me.*

► *Edgar Rice Burroughs, A Princess of Mars*

---

# Preface

## Why a New Book on XSLT?

XSLT, or Extensible Stylesheet Language for Transformations, has been around as a specification since late 1999, and the first books about it started appearing not long after. The most well-known texts arrived in the early 2000s, providing guidance for many an XSLT novice in the duration. With the publication of XSLT 2.0, new editions arrived, and then, for a long time, the land of XSLT books has been fairly quiet.

As one of those XSLT novices back in 2004, I found that the early XSLT documentation spent much of its bulk describing elements of the language in a tone that suggested deep familiarity with the specification, but much less interest in the problems that XSLT might solve. As I dealt with the requirements for the company I worked with, I struggled to assemble the bits and pieces of XSLT into larger solutions. It was like being given a bucket of electronic components and being asked to assemble a radio without a schematic. I didn't need academic discourses, I needed to see how to apply it in real life.

Then I met other programmers who told me about similar experiences with XSLT's learning curve. An accomplished documentation toolsmith I worked with used to lament that it would be nice to have a book that showed XSLT in action on real-world problems. A college-level computer science teacher talked about the difficulty he had when he first encountered XSLT. If these folks were frustrated, pity the poor student!

Their comments made me wonder: what makes XSLT so difficult to learn?

I don't know that there's a single good answer, but here are a few guesses:

- An uncommon, verbose syntax
- The relative immaturity of the language when most of the books appeared
- A small range of applications, meaning a small audience, meaning a small support community and thin documentation.

In reflecting on these issues, it seemed to me that a book centered around the concepts involved in solving certain basic problems could go a long way toward helping people get over the big bump of the initial XSLT learning curve.

## A Little about Me

I can't claim that knowledge of XSLT is my primary qualification for writing this book. As an XSLT programmer, there are a lot sharper tools in the shed! What I bring to the task is a career as a technical writer/documentation manager and software tools user/developer. If there's one thing I enjoy, it's taking complex information and making it accessible to other people. One of the most gratifying comments I hear after someone reads something I've written is, "I thought it was hard, but there's really nothing to it."

I've used XSLT for over 10 years, and I've solved some thorny problems along the way. I've used XSLT to deconstruct the DITA Open Toolkit, which turned out to be an excellent training ground (although I wouldn't recommend it to the faint of heart). Other projects took me through a conversion of markup for ancient Greek manuscripts, timesheet applications, NROFF and Wiki markup conversions, and a variety of conversions to PDF and ebook formats. Working on these complex applications taught me that XSLT is a powerful, flexible solution for manipulating XML. Once I got past the initial conceptual barriers, I found it to be a fascinating and extremely useful language.

Recently, I spent a few years working for The Pragmatic Programmers, helping with their publishing system and learning their writing methods. I have tailored this book to fit the style of books from Pragmatic Bookshelf: friendly, encouraging, and moving gently along the path from initiation toward mastery.

## A Little about You!

You may be a student or a young professional who wants to tackle the wide world of XML processing, and you are looking for a "leg up." On the other hand, you may be a seasoned pro who has dealt with XML before, but you've used other tools and have found them to be less than adequate. In both cases, you've been wondering whether XSLT is a better solution—and then you've gotten stuck on its syntax. (To quote Dave Thomas in conversation, "XSLT is a wonderful functional language with a really lousy syntax.")

One thing I do know is that you'll need a few skills to get started:

- Knowledge of the major components of XML—elements (tags), attributes, comments, and so forth. Nothing exotic.

- A few basic principles of computer programming logic
- Low-level familiarity with HTML (to understand the outputs created in the examples)

You may find this book starts a little slow, although we'll get started with some hands-on examples pretty quickly. First we'll lay some groundwork, covering a lot of ground in a short time. Once we get past the conceptual introduction, there will be a good bit of juggling between describing components and showing how they work in the context of a solutions. While I present most of the code for the examples in full, I'm hoping that you'll be trying out the code yourself, working with XML examples that you process in a web browser or with a standalone XSLT processor.

And don't stop with just the examples provided in this book! I've found there's nothing more effective for learning XSLT than experimenting with it. I'm hoping you will use the examples in this book as jumping-off points for your own explorations.

So, regardless of your level of experience, I hope you'll bring curiosity and a sense of adventure to this book...and even a sense of fun as you play with the possibilities.

## What You'll Need for This Book

The requirements for creating and processing XSLT are fairly simple: you can do it with as little as a text editor and a web browser. However, there are benefits to using more robust tools. The good news is that there are a lot of robust tools available at no charge.

Here's the list of what we'll use in this book:

- An XML editor, preferable one that understands the XSLT syntax
- A web browser that supports XSLT processing
- A standalone XSLT processor

I'll be recommending specific packages later in this book—we'll need to discuss what options are available, since there are many possibilities. At the moment, though, let's not worry about it—the requirements are fairly simple and will be easy to put in place when we need them.

And that's it! We'll be sticking with XSLT 1.0 for the most part, since that's what is supported by web browsers. It should be fairly easy to find tools that support the examples in this book.